

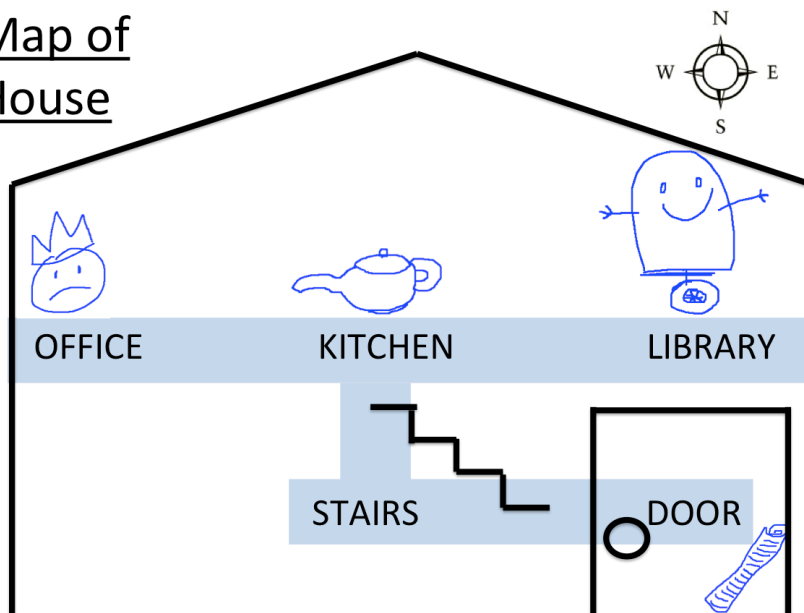
A Robot Maid

Two files are used in this example: `Robot.java` and `TestRobot.java`. `TestRobot.java` stores all the code that runs the overall program while `Robot.java` stores all the logic. Let's see what `Robot.java` does.

A robot lives in a house with a master. Generally, the robot cleans up in the LIBRARY room, while the master rests in his OFFICE. Once in awhile, the master calls for the robot, and either asks for tea, newspaper, or for no good reason at all. So the robot roams around the house to accomplish what his master wants and goes back to cleaning.

A View of the Robot's World

Map of House



How To Navigate Around

- There are 4 directions: NORTH, EAST, SOUTH, WEST
- There are 5 locations: OFFICE, KITCHEN, LIBRARY, STAIRS, DOOR
- When the robot is in a particular location, it can move in a direction, and either:
 - Hit the wall (ACK!) or
 - Successfully move to another room

Getting Started

`TestRobot.java` starts the program:

- Creates a new Robot object called “roomba”
- Lets the master call it into the OFFICE

Follow the code in `Robot.java` to continue the flow of the program:

- `comeHere` method:
 - If the robot is somewhere other than the OFFICE, navigate to the OFFICE
 - Display the menu of options for master to choose from
 - Process master’s command

Demo – you are the master! Which command do you want the robot to do?

Additional logic in `Robot.java`:

- `getPaper` method:
Tells the robot to go to the DOOR, the robot picks up the newspaper, delivers it to the master in the OFFICE, and goes back to the LIBRARY
- `getTea` method:
Tells the robot to go to the KITCHEN, the robot makes tea, delivers it to the master in the OFFICE, and goes back to the LIBRARY
- `goBack` method:
Tells the robot to go back to the LIBRARY
- `gotoLoc` method:
Given a specific location, generate a *random* direction, move to it!

What You Have To Do

Complete the `possibleMoves` method by explaining the logic behind the set of possible moves available to the robot.

That is, for each possible location that the robot can be at (there are 5 in total), list all possible directions that it can move in and where the robot ends up after moving in that direction.

Complete the blanks in the code on the following page based on the map layout of the house. Some examples are provided within the code for you to follow.

When you are done, copy the code into the program that has been provided to you. You will need to copy this code into the `possibleMoves` method exactly the way it is typed out (be sure to maintain the same capitalization and punctuation).

Coding Worksheet for Robot Maid

```
if( currLoc == OFFICE )
{
    if( dir == EAST ) currLoc = KITCHEN;
}

if( currLoc == KITCHEN )
{
    if( dir == _____ ) currLoc = _____;
    if( dir == _____ ) currLoc = _____;
    if( dir == _____ ) currLoc = _____;
}

if( currLoc == LIBRARY )
{
    if( dir == _____ ) currLoc = _____;
}

if( currLoc == _____ )
{
    if( dir == _____ ) currLoc = _____;
    if( dir == _____ ) currLoc = _____;
}

if( currLoc == _____ )
{
    if( dir == _____ ) currLoc = _____;
}
```

Common Pitfalls To Avoid

- Remember to spell out the directions and locations exactly the way it was presented on the previous page
- The possible moves are defined based on the layout of the map on the previous page
- You only need to specify what results in a successful move; anything *not* specified in your code will be treated as a failed move (and the robot hits the wall)